

# 东信银星凭条打印机接口

## 调用方式

POST, 服务端口 8082

样例: http://localhost:8082/Init

返回报文retCode >= 0, 如果没有特殊说明则为成功 < 0则为失败, retMsg为具体信息

## 修改记录

2018-03-22

条码节点增加 BarcodeType , 条码类型

## 接口定义

### 一、初始化 Init

报文样例:

通过配置文件读取参数-如需修改参数,直接修改配置文件中的参数即可

配置文件路径config\\conf\_ptr.json

conf\_ptr.json如下

```
{
  "port": "USB",
  "rate": 9600
}
```

Port 为具体串口号或者USB 两种根据具体物理场景来写

rate 为凭条打印机的波特率 一般为:9600 19200 38400

port为USB时 rate无效, 默认9600即可

返回样例:

成功:

```
{
  "retCode": 1,
  "retMsg": "成功"
}
```

失败:

失败 例子1

```
{
  "retCode": -1,
  "retMsg": "未指定的错误"
}
```

失败: 例子2

```
{
  "retCode": -9000,
  "retMsg": "凭条打印机打开失败"
}
```

### 二、查询状态 Status

报文样例:

参数:

无

返回样例:

```
成功
{
  "retCode": 1,
  "retMsg": "成功",
  "detail": {
    "PaperStatus": "正常",
    "PaperCode": 0,
    "CutStatus": "正常",
    "CutCode": 0,
    "CoverStatus": "合上",
    "CoverCode": 0
  }
}
失败:
{
  "retCode": -9013,
  "retMsg": "凭条打印机未初始化"
}
状态1
{
  "retCode": 1,
  "retMsg": "成功",
  "detail": {
    "PaperStatus": "纸将尽",
    "PaperCode": -1,
    "CutStatus": "正常",
    "CutCode": 0,
    "CoverStatus": "合上",
    "CoverCode": 0
  }
}
状态2
{
  "retCode": 1,
  "retMsg": "成功",
  "detail": {
    "PaperStatus": "缺纸",
    "PaperCode": -2,
    "CutStatus": "正常",
    "CutCode": 0,
    "CoverStatus": "正常",
    "CoverCode": 0
  }
}
状态3
{
  "retCode": 1,
  "retMsg": "成功",
  "detail": {
    "PaperStatus": "缺纸",
    "PaperCode": -2,
    "CutStatus": "正常",
    "CutCode": 0,
    "CoverStatus": "上盖打开",
    "CoverCode": -1
  }
}
```

### 三、打印文本或条码Textout

Textout内部调用了设置纸张标准模式,清空缓冲区, 以及打印和切纸4个功能.

#### 报文样例:

#### 参数:

```
{
  "PrintData": [
  {
    "Type": "Text",      Type为要打印的类型 Text-文本(中文,数字,字符等)或Barcode-条码
    "Data": "你我我好他也好",  Data为需要打印的数据
    "X": 5,              X为打印条码的起始横坐标0-65535 可不传此项,默认为5
    "Y": 5,              Y为打印条码的起始纵坐标0-65535 可不传此项,默认为5
    "WidthTime": 1,      横向放大倍数, 1-6 可不传此项,默认为1
    "HightTime": 1,      纵向放大倍数, 1-6 可不传此项,默认为1
    "nFontType": 0,      字体类型风格, 0-3 可不传此项,默认为0
    "FontStyle": 0       字体风格, 0-1F88 可不传此项,默认为0x00
  },
  {
    "Type": "Barcode",    Type为要打印的类型 Text-文本(中文,数字,字符等)或
    "Data": "12345678901", Data为需要打印的数据 条码现在只支持11或者12位数字
    "BarcodeType": 0x48   条码类型, 取值范围
                          TKIOSK_BARCODE_TYPE_UPC_A  0x41  //UPC-A
                          TKIOSK_BARCODE_TYPE_UPC_E  0x42  //UPC-E
                          TKIOSK_BARCODE_TYPE_JAN13  0x43  //JAN13(EAN13)
                          TKIOSK_BARCODE_TYPE_JAN8   0x44  //JAN8(EAN8)
                          TKIOSK_BARCODE_TYPE_CODE39  0x45  //CODE39
                          TKIOSK_BARCODE_TYPE_ITF    0x46  //ITF
                          TKIOSK_BARCODE_TYPE_CODEBAR 0x47  //CODEBAR
                          TKIOSK_BARCODE_TYPE_CODE93  0x48  // CODE93 (默认)
                          TKIOSK_BARCODE_TYPE_CODE128 0x49  //CODE 128
  }
  ],
  "CutPaper": [
  {
    "Type": "CutPaper",  Type为需要切纸的类型 CutPaper-全切 或 HalfCut-半切
    "Data": 0            Data为 0-255 默认传0
  }
  ]
}
```

备注  
Textout样例报文中 PrintData数组里面可以有多个文本或者条码类型,即可以打印多个条码和文本  
CutPaper只能是一个

例如:

```
{
  "PrintData": [
  {
    "Type": "Text",
    "Data": "你我我好他也好"
  },
  {
    "Type": "Text",
    "Data": "你我我好他也好"
  },
  {
    "Type": "Barcode",
    "Data": "12345678901"
  },
  {
    "Type": "Barcode",
    "Data": "12345678901"
  }
  ]
}
```

```
},
{
  "Type": "Text",
  "Data": "你我我好他也好"
}
],
"CutPaper": [
{
  "Type": "CutPaper",
  "Data": 0
}
]
}
```

返回样例：

```
成功
{
  "retCode": 1,
  "retMsg": "成功"
}
失败
{
  "retCode": -9013,
  "retMsg": "凭条打印机未初始化"
}
```

---

## 四、清空缓冲区内容

ClearBuffer(Textout js里面是有封装调用此方法的)

报文样例：

参数：

无

返回样例：

```
成功
{
  "retCode": 1,
  "retMsg": "成功"
}
失败
{
  "retCode": -9013,
  "retMsg": "凭条打印机未初始化"
}
```

---

## 五、复位打印机 Reset

报文样例：

参数：

无

返回样例：

```
成功
{
  "retCode": 1,
  "retMsg": "成功"
}
失败
{
  "retCode": -9013,
  "retMsg": "凭条打印机未初始化"
}
```

---

## 六、设置字符行高 SetLineSpacing

报文样例：

参数：

```
{
  "LineHeight": 1
}
```

LineHeight 为字符行高,范围:0-255    可不传此参数,默认为1

返回样例：

```
{
  "retCode": 1,
  "retMsg": "成功"
}
```

---

## 七、关闭凭条打印机 Close

报文样例：

参数：

```
无
```

返回样例：

```
{
  "retCode": 1,
  "retMsg": "成功"
}
```